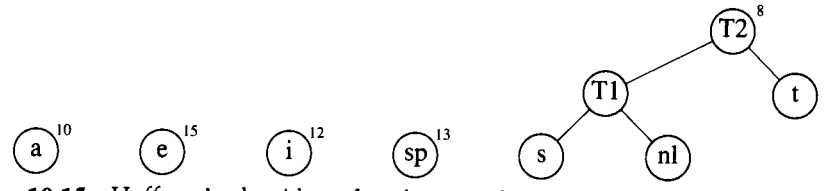

Figure 10.15 Huffman's algorithm after the second merge

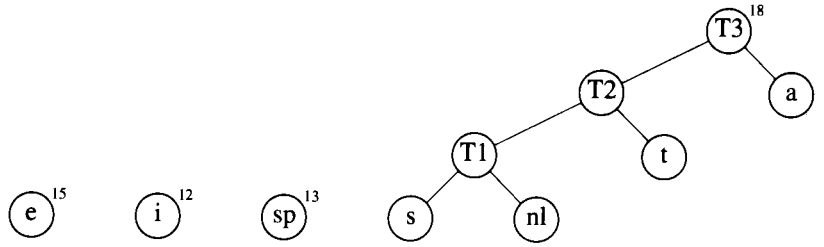

Figure 10.16 Huffman's algorithm after the third merge

old trees, and can thus be easily computed. It is also a simple matter to create the new tree, since we merely need to get a new node, set the left and right pointers, and record the weight.

Now there are six trees, and we again select the two trees of smallest weight. These happen to be $T1$ and t, which are then merged into a new tree with root $T2$ and weight 8. This is shown in Figure 10.15. The third step merges $T2$ and a, creating $T3$, with weight $10 + 8 = 18$. Figure 10.16 shows the result of this operation.

After the third *merge* is completed, the two trees of lowest weight are the single-node trees representing i and the blank space. Figure 10.17 shows how these trees are merged into the new tree with root $T4$. The fifth step is to merge the trees with roots e and $T3$, since these trees have the two smallest weights. The result of this step is shown in Figure 10.18.

Finally, the optimal tree, which was shown in Figure 10.11, is obtained by merging the two remaining trees. Figure 10.19 shows this optimal tree, with root $T6$.

Figure 10.17 Huffman's algorithm after the fourth merge